teaching_python

# 10+ years of successes, trials and tribulations

Bruce Fuda
@Bruce1979
The guy at the front...

Bruce Fuda
Director of Technologies

experience = [“teacher”: “12 years; government high schools and colleges, ACT”,
“developer”: “web applications - school support; python/flask, HTML/CSS/JS”
“curriculum”: “adviser and writer for Australian Curriculum: Technologies”]
the_educational_landscape

# the need for a robust, engaging and rigorous Computer Science curriculum
It’s a bit of a mess…

- Hasn’t kept up with changes in industry
- Dated curriculum
- Lack of teacher expertise
- Inconsistent between jurisdictions; schools
- Low demand from students and parents
- Has an image problem
- But… the Australian Curriculum has now been endorsed!

http://theday.co.uk/technology/boring-computer-curriculum-just-doesn-t-get-it, 12 January 2012

teaching_python
Bruce Fuda
The Australian Curriculum:

Digital Technologies

- **Interactions and Impact**
  People, systems, processes, sustainability, empowerment

- **Digital Systems**
  Hardware, software, networks and the Internet

- **Data**
  Properties, sources, collection, symbolism, patterns, contexts

- **Specification, Algorithms, Interpretation**

- **Data Collection, Representation, Interpretation**

- **Abstraction**
  Underpins all other concepts

- **Algorithms**
  Defining problems and designing solutions

Bruce Fuda

Python in Education Seminar
The environment is changing…

**Implementation**

Since endorsement, authorities nationwide have committed to implementation of the Australian Curriculum in full from Foundation to Year 10.
my_experience

# 2005-2016: deciding on a language and approach to teaching programming
Teaching Programming isn’t easy...
Hard
Compared to other subjects due to lack of familiarity; opt out in high stakes situations e.g. Year 11/12

Unfamiliar
Programming is new to almost all students when it is introduced at school
Experience
IT Classes are boring - applications, process;
Would rather do something stimulating

Expectation
Most parents and students don’t know what to expect;
Using an iPad != capable developer
The dark times - before Python…

**Starting Out**

---

**Day 0**

STROMLO HIGH SCHOOL, ACT  
2005

**Javascript**

Only programming at the school was in an *Internets and Intranets* unit; Javascript quizzes in the web browser; substitute values into existing code

---

**Internets and Intranets**

expanded to include server-side programming through PHP; students largely relied on re-using code from the Internet

---

STROMLO HIGH SCHOOL, ACT  
2006

**PHP**

---

STROMLO HIGH SCHOOL, ACT  
2007

**Java**

By 2008, an OO course had been written using Java as the main language; overly complex for introducing basics; only appealed to top students
Python Adoption

The period of enlightenment - Python

Learn to program with Python through browser; high engagement; well scaffolded

STROMLO HIGH SCHOOL, ACT 2009

The NCSS Challenge

InTEACT, ACT 2012

Python Growth

Successes from NCSS Challenge shared with teachers across ACT; Increased participation; Higher visibility of Python in high schools

Gungahlin College uses Python as primary language (some Arduino); Highest IT enrolments in the ACT

GUNGAHLIN COLLEGE, ACT 2016

Python Dominance

Now
Python strengths

01 Simplicity
- Installation and use
  - No special IDE necessary
  - Cross-platform
  - Very little technical support required

02 Friendly
- Clean, readable syntax
  - Statements, syntax and grammar are easy for beginning programmers

03 Powerful
- Scripting, Object-Oriented
  - Built-in and readily available modules make doing interesting things a breeze

04 Flexible
- Beyond basics
  - With support of frameworks (such as Flask and plot.ly) it can be used in wider applications

05 Resources
- Learn what you want
  - Lots of online resources to teach Python, many specifically targeted to education

06 Support
- You’re not alone
  - The Python community is active and supports beginners (teachers and students)
Obstacles to learning

- Graphics and games
- Peripherals
- Bundling applications

… but experience solves a lot of this!
lessons_learned

# avoiding the traps; exploiting the strengths
Make it as painless as possible:

Fundamentals

- assignment
- comparison
- control structures
- functions definition
- class definition
interactivepython.org

How to Think Like a Computer Scientist
Think; code

think about the problem
design the solution
identify the steps

then write your code
Define problem

Rapid research

Implement

Test / feedback

Ideate
Relevance

meaningful tasks
appropriate complexity
solve real problems
Find out about some of these student projects and others in Sunday’s Student Showcase session at 1:40pm
Find out about some of these student projects and others in Sunday's Student Showcase session at 1:40pm
Find out about some of these student projects and others in Sunday’s Student Showcase session at 1:40pm.
Find out about some of these student projects and others in Sunday's Student Showcase session at 1:40pm.
Welcome to Prophecy by GXT
Copyright 2015

For Predictive Text, enter PT.
For Parody Generator, enter PG
To Edit Dictionary, enter ED.
For more options, enter M.
To reset dictionary, enter RESET
To end program, enter end.
Enter Choice: PG

Starting word: Harry
Length of text: 300

Harry and escape and make up into an excessively interested in the gps and the submissive with the crowd into my mouth presses tobias crouches in here he can look at each other weekend? It is spacious room. It was cool smooth back with hovercraft at the corner the school where they were safe. It is making amends and smile that's because it was ever agree to the deadbolt. It happened before I might have put poison in. The lunchtime wouldn't settle into that day to get for me and I burst through me. A boy's a noise they'd tear with you for being slow motion the time we knock on to do you tonight? The image of all into the mood shifted suddenly as he deliberately unhurried no better that only they need a full of me tonight I knew charlie I close to seeing him this evening. A document. It seemed almost breaks down to do this tracker jackers. A miserable said she found our way and i'm not comprehending. A whisper pressing hard limits no one for a few more distinct dent in mind it's just to the last night. It is. It isn't your own and I quite a wizard coins from the house I'm all over the parachutes? The big day. It through; he lifted a riskfree environment in the probable than a large lobby waiting for casual erect the point plane and then whether they march day might be brave and flew up and take my father cronin moving and derricks in their desire to do as well. It down I won't. A little finger pressing myself that someday you really think of which might make with her? A good tough but I wish we were sitting. It was relatively tasteless but aware you

Parody generating finished, returning to menu.
Please enter another text: https://en.wikipedia.org/wiki/Python_(programming_language)

# Summerized Text #

(\text{Python}, \text{the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all of its variant implementations.})

[42] \text{Python uses dynamic typing and a mix of reference counting and a cycle-detecting garbage collector for memory management.}

Methods on objects are functions attached to the object's class; the syntax \text{instance.method(argument)} is, for normal methods and functions, syntactic sugar for \text{Class.method(instance, argument)}.

[72] \text{Python allows boolean expressions with multiple equality relations in a manner that is consistent with general use in mathematics.}

\text{Python has a large standard library, commonly cited as one of Python's greatest strengths, [77] providing tools suited to many tasks.}

The \text{Nokia N900} also supports Python with GTK widget libraries, with the feature that programs can be both written and run on the target device.

[98] For example, the metasyntactic variables often used in Python literature are spam and eggs, instead of the traditional foo and bar.

# End of Text #
Find out about some of these student projects and others in Sunday's Student Showcase session at 1:40pm
Collaborate

make it social

differentiate on interest

draw on experiences
Open-ended

- start with something concrete
- provide a progression
- leave the destination undefined
Ensure that there’s something for everyone

**IT Task Structure**

**Document**
- Document existing code
- Understand the role of each part

**Improve**
- Fix any broken features
- Add new (similar) ones

**Expand**
- Increase the scope
- Add new, unique features

**Redefine**
- Alter the underlying structure
- Show us what you can do!

We can all do this with teacher support

We’re all challenged by this

*teaching_python*
Bruce Fuda
reflection

# the key to iterative improvement
I asked my students…

**Why did you like learning Python?**

“…since it was pretty intuitive and close to natural language, it was easy to follow, and I tend to like anything that I can understand…”

Kat Gill  
Student  
Gungahlin College  
New programmer

“It not only taught me how to write code but also… how to think logically about the problems I was trying to solve… making it much easier for me in the future to learn other programming languages.”

Damanvir Singh  
Student  
Gungahlin College  
5 years experience

---

teaching_python  
Bruce Fuda
The tl;dr version

Success checklist

1. **Teach the fundamentals**
   Make sure students are equipped with the tools they need to solve problems

2. **Emphasise thinking**
   Writing the code is the easy part if your solution is well designed

3. **Keep things interesting**
   Make tasks genuine; provide context and a reason for writing a program

4. **Everyone is on the same team**
   Rivalry is a part of school - make it friendly and encouraging, not antagonistic

5. **Don’t restrict creativity**
   Programming is as much an art as it is a tool
Thanks for your attention

Questions?

bruce.fuda
@Bruce1979
+BruceFuda
bruce@fuda.me
bruce-fuda
http://fuda.me/